## LISTING OF THE CLAIMS

At the time of the Action:

Pending Claims: 1-5, 7-8, and 10-34

Canceled Claims: 6 and 9

After this Response:

Pending Claims: 1-5, 7-8, and 10-34

Amended Claims: 1-5, 7-8 and 10-34

Canceled Claims: 6 and 9

1. (Currently Amended) A method of file system protection for a resource-sparing operating system image, comprising:

loading, with a client computing device, a first ~~resource-sparing operating system~~ image of the resource-sparing operating system (OS) that includes processor instructions into random access memory (RAM), the first image including an embedded second image of a catalog file comprising client device attributes ~~embedded therein~~;

creating, with the client computing device, a first hash of the first image;

extracting with the client computing device a second hash ~~of~~ from the second image ~~from~~ of the catalog file;

comparing with the client computing device the first hash and the second hash; and

~~blocking the use of the image to boot the computing device if the first hash and the second hash do not match, or~~

validating with the client computing device the use of the first image to boot the computing device if the first hash and the second hash match.

2. (Currently Amended) The method of claim 1, wherein the comparing includes determining that the first hash and the second hash do not match, and wherein further the blocking the use of the first image to boot the computing device if the first hash and the second hash do not match comprises determining an operational mode of the computing device is set to a run mode of operation.

3. (Currently Amended) The method of claim 1, wherein the comparing includes determining that the first hash and the second hash match, and wherein further the validating the use of the first image to boot the computing device if the first hash and the second hash match further includes validating a signature certification of the first image.

4. (Currently Amended) The method of claim 1, further comprising evaluating a signature certification of the catalog file in the second image to determine if the signature certification of the catalog file is valid, and wherein the blocking the use of the first image to boot the computing device if the first hash and the second hash do not match further includes blocking the use of the first image to boot the computing device if the signature certification of the catalog file cannot be validated.

5. (Currently Amended) The method of claim 4, further comprising determining whether an operational mode of the computing device is set to a run mode of operation.

6. (Canceled)

7. (Currently Amended) The method of claim 1, further comprising extracting first make and model attributes from the second image of the catalog file, comparing the first make and model attributes from the second image of the catalog file with second make and model attributes of the computing device, and wherein the blocking the use of the first image to boot the computing device if the first hash and the second hash do not match further includes blocking the use of the first image to boot the computing device if the first make and model attributes do not match the second make and model attributes.

8. (Currently Amended) The method of claim 7, further comprising determining whether an operational mode of the computing device is set to a run mode of operation.

9. (Canceled)

10. (Currently Amended) The method of claim 1, further comprising booting the computing device from a third ~~prior~~ image of a prior resource-sparing OS already loaded in flash memory of the computing device.

11. (Currently Amended) A method of file system protection for a resource-sparing operating system (OS) image, the resource-sparing OS image including a catalog file image embedded therein, the method comprising:

~~examining the~~ comparing information extracted from the embedded catalog file image ~~and~~ with information obtained from the resource-sparing OS image to determine if the resource-sparing OS image is a properly released resource-sparing OS image; and

blocking use of the resource-sparing OS image to boot the computing device when the examining determines that the resource-sparing OS image is not a properly released resource-sparing OS image.

12. (Currently Amended) The method of claim 11, wherein the ~~examining~~ comparing is initiated upon a request to update the resource-sparing OS image, the method further including loading an update to the resource-sparing OS image into random access memory (RAM) when the examining determines that the resource-sparing OS image is a properly released resource-sparing OS image.

13. (Currently Amended) The method of claim 11, wherein the ~~examining~~ comparing is initiated upon a reset of the computing device.

14. (Currently Amended) The method of claim 11, wherein the ~~examining~~ comparing information extracted from the embedded catalog file image with information obtained from the resource-sparing OS image to determine if the resource-sparing OS image is a properly released resource-sparing OS image comprises:

creating a first hash of the resource-sparing OS image;

extracting a second hash ~~of the image~~ from the catalog file image; ~~and~~

comparing the first hash and the second hash~~.~~; and

~~wherein~~ indicating ~~a mismatch provides an indication~~ that the resource-sparing OS image is not a properly released resource-sparing OS image upon a mismatch between the first hash and the second hash.

15. (Currently Amended) The method of claim 14, further comprising determining an operational mode of the computing device.

16. (Currently Amended) The method of claim 11, wherein the catalog file image is an image of a catalog file, and wherein the ~~examining~~ comparing information extracted from the embedded catalog file image with information obtained from the resource-sparing OS image to determine if the resource-sparing OS image is a properly released resource-sparing OS image comprises:

extracting a signature certification from the catalog file;

validating the signature certification; and

wherein failure of the act of validating the signature certification provides an indication that the resource-sparing OS image is not a properly released image.

17. (Currently Amended) The method of claim 16, further comprising determining an operational mode of the computing device.

18. (Currently Amended) The method of claim 11, wherein the catalog file image is an image of a catalog file, and wherein the comparing information extracted from the embedded catalog file image with information obtained from the resource-sparing OS image to determine if the resource-sparing OS image is a properly released resource-sparing OS image ~~examining~~ comprises:

extracting first make and model attributes from the catalog file;

comparing first make and model attributes from the catalog file to second make and model attributes of the device; and

wherein a mismatch between the first and the second make and model attributes provides an indication that the resource-sparing OS image is not a properly released image for the device.

19. (Currently Amended) The method of claim ~~18~~1, further comprising blocking with the client computing device the use of the first image to boot the computing device if the first hash and the second hash do not match~~determining an operational mode of the device~~.

20. (Currently Amended) The method of claim 11, further comprising loading the resource-sparing OS image into random access memory (RAM) of the device, and wherein the ~~examining~~ comparing is processed after the loading.

21. (Currently Amended) The method of claim 11, wherein if the ~~examining~~ comparing determines that the resource-sparing OS image is a properly released image, the method further comprising:

erasing a previous resource-sparing OS image from flash memory of the computing device;

programming the flash memory of the computing device with the properly released resource-sparing OS image.

22. (Currently Amended) The method of claim 11, wherein the catalog file image is an image of a catalog file, and wherein the comparing information extracted from the embedded catalog file image with information obtained from the resource-sparing OS image to determine if the resource-sparing OS image is a properly released resource-sparing OS image ~~examining~~ comprises:

creating a first hash of the resource-sparing OS image;

extracting a second hash ~~of~~ from the ~~image from the~~ catalog file image;

comparing the first hash and the second hash;

extracting a signature certification from the catalog file;

validating the signature certification; and

extracting first make and model attributes from the catalog file;

comparing first make and model attributes from the catalog file to second make and model attributes of the device; and

indicating that the resource-sparing OS image is not a properly released image for the device ~~wherein~~ upon any one of a first mismatch between the first hash and the second hash, a failure of the act of validating the signature certification, and a second mismatch between the first and the second make and model attributes ~~provides an indication that the image is not a properly released image for the device~~.

23. (Currently Amended) A portable computing device, comprising:

flash memory, the flash memory including a protected area and an unprotected area;

a bootloader stored in the protected area of flash memory, the bootloader containing a cryptographic module;

an operating system (OS) image ~~stored~~ installed in the unprotected area of flash memory;

random access memory (RAM); and

wherein the cryptographic module of the bootloader is operative to examine an update image to the OS image ~~update~~ to determine if the update image ~~update~~ should be programmed into the unprotected area of flash memory to boot the computing device, wherein a signed catalog image is an image of a signed catalog file and is embedded in the update image, wherein the signed catalog file is derived by signing a catalog file, and wherein the cryptographic module is operative to program the update image into the unprotected area of flash memory boot the computing device based on a determined relationship between information ~~included~~ extracted from~~in~~ ~~a~~ the embedded signed catalog file ~~embedded in the image update~~and one of information about the components of the computing device and information determined from the update image.

24. (Currently Amended) The device of claim 23, wherein the cryptographic module programs the update image ~~update~~ into the unprotected area of flash memory when the device is in test mode.

25. (Currently Amended) The device of claim 23, wherein the bootloader stores the update image ~~update~~ in the RAM until the cryptographic module determines that the update image ~~update~~ should be programmed into the unprotected area of flash memory to boot the device.

26. (Currently Amended) The device of claim 25, wherein the cryptographic module calculates a first hash of the update image ~~update~~, extracts a second hash from the catalog file image, and compares the first hash and the second hash, the cryptographic module blocking use of the ~~image~~ update image when the first hash and the second hash do not match.

27. (Currently Amended) The device of claim 25, wherein the cryptographic module extracts a signature certification from the catalog file and attempts to validate the signature certification, the cryptographic module blocking use of the OS image update when the signature certification cannot be validated.

28. (Currently Amended) The device of claim 25, wherein the operating system (OS) image is an image of an operating system, wherein the update image is an image of an update file to the OS, wherein the cryptographic module is operative to program the update image into the unprotected area of flash memory boot the computing device based on a determined relationship between information extracted from the signed catalog file and information about the components of the computing device by:

~~extracts~~ extracting make and model attributes from the catalog file and ~~compares~~ comparing them to make and model information for the computing device, and ~~the crypto module~~

blocking use of the ~~image~~ update image when the make and model attributes of the catalog file ~~image update~~ do not match the make and model attributes of the computing device.

29. (Currently Amended) The device of claim 25, wherein the bootloader erases a current device image from the unprotected area of flash memory and programs the update image ~~update~~ into the unprotected area of flash memory when the cryptographic modules determines that the ~~image~~ update image may be used to boot the device.

30. (Currently Amended) The device of claim 23, wherein a second cryptographic module of a Mira shell is operative upon a reset of the computing device to examine the installed operating system image to determine if the installed operating system image should be used to boot the computing device based on information included in ~~a~~ the signed catalog file ~~embedded in the installed operating system image~~.

31. (Currently Amended) The device of computing claim 30, wherein the cryptographic module in the Mira shell calculates a first hash of the installed operating system image, extracts a second hash from the catalog ~~file~~image, and compares the first hash and the second hash, the cryptographic module in the Mira shell blocking use of the update image ~~update~~ when the first hash and the second hash do not match.

32. (Currently Amended) The device of computing claim 30, wherein the cryptographic module in the Mira shell extracts a signature certification from the catalog file and attempts to validate the signature certification, the cryptographic module in the Mira shell blocking use of the installed operating system image when the signature certification cannot be validated.

33. (Currently Amended) The computing device of claim 30, wherein the cryptographic module in the Mira shell extracts make and model attributes from the catalog file and compares them to make and model information for the computing device, wherein the operating system image is an image of an operating system for the computing device, the cryptographic module in the Mira shell blocking use of the installed operating system image when the make and model attributes of the installed operating system ~~image~~ do not match the make and model attributes of the computing device.

34. (Currently Amended) The computing device of claim 30, wherein the cryptographic module in the Mira shell allows the installed operating system image to be used to boot the computing device when the computing device is in test mode.